

Clock Events Diagnostic

Page application

Thu, May 11, 2000

This page application is a tool to diagnose clock event occurrences as measured in an IRM. Each event causes an interrupt whose service routine reads from a FIFO the event or events causing the interrupt. A new option has been added so that a data stream can be written containing the sequential occurrences of clock events. The page application reads from the data stream queue to obtain the necessary raw data to produce the listings of event occurrences.

A user can specify a triggering clock event to be used in building the listing. A data request is made for the event records in the queue. When a record for that clock event is found, a table is built of all event occurrences from that time until the buffer is filled. The records are then processed to produce a serial output, with the screen being used to show the first part of it.

One listing shows on each line the clock event, the relative time since the triggering clock event, and the delta time between the last two such clock events. Every time the triggering clock event occurs, the relative times begin again.

The encoded listing is produced as the events are read out of the data stream queue. The data request asks for 15Hz replies, and each record is processed for serial output. This continues until the user "clicks" again to stop it, and the request is canceled.

The event detection logic has control bits that can ignore certain fast clock events, so that the total event rate is more manageable. One that is traditionally omitted is event 07, the 720Hz event.

Interrupt on the node# row to begin the data collection. The display updates the count of event records received. Each record is written to the serial output according to the processing logic. One may specify that some events are to be skipped and not output. One may also specify that only certain events are to be included in the listing output. Interrupt again to stop the data collection and listing generation.

Interrupt on the EVTR row to enter changed selection values, and also to repeat the listing using the indicated selection criteria. The same data that was captured from the interrupt on the NODE row is reprocessed, and a new listing is generated.

The 32-character page display format is as follows:

```
01234567890123456789012345678901
Q CLK EVENT TIMES 05/11/00 0908
NODE<0509> #EV=      64 LIST<0576>
EVTR=11  +  .. .. .. .. ..
11      0.000
0C      0.005
22      5.425
3F      6.426
7A     24.488
18     38.004
0F     49.730
11      0.000      66.657
0C      0.007      66.659
8F      8.992
7A     10.573      52.742
18     38.001      66.654
```

The event number following the EVTR prompt specifies the triggering event number. Events are ignored until this event is seen, then the listing generation is enabled with the relative times being referenced to the time stamp of the triggering event. If this same triggering event recurs, the relative times start over. The delta time values only relate to the events that have been seen during this processing. The first occurrence of any given event will therefore not have a delta time value.

The + sign means that the following events are to be included in the listing, all others being omitted. If a – sign is used, the following events are to be omitted from the listing, which is otherwise complete.

The times are in units of milliseconds. Note that time stamps for events that ideally occur simultaneously may show a 5 microsecond separation, rather than only the one microsecond that is demanded by the serial format. This is because the interrupt routine software that processes the interrupts takes some time. It reads out successive events from a hardware FIFO upon the occurrence of an interrupt. The loop time seems to be about 5 microseconds in the 68040-based IRM.

The NODE text lights up during data collection. The number following this can count to one million. Alternatively, the count could be set to indicate only the number of listing lines produced. The character following the node number field is used to indicate the status of the reply data.

The specification of up to 7 events for selections includes limited wildcarding. One may specify nX events or Xn events, each X indicating a "don't care" for one nibble. One may use XX, but it is not very interesting, as it would be the default case with no events specified. Internally, an array of 2-byte values specifies the selection-by-event parameters. The hi byte is 00, 0F, F0, or FF, according to the cases nn, nX, Xn, and XX. The lo byte holds the event number.